INTERNATIONAL STANDARD

# ISO 28640

First edition
2010-03-15

## Random variate generation methods

*Méthodes de génération de nombres pseudo-aléatoires*



Reference number
ISO 28640:2010(E)

## Contents

Page

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 28640 was prepared by Technical Committee ISO/TC 69, *Applications of statistical methods*.

This is the first edition.



 

## Introduction

This International Standard specifies typical algorithms by which the users can regard the generated numerical sequences as if they were real random variates.

Nowadays most statisticians, scientists and engineers have enough computer power at their disposal to carry out large computer simulations, and it is important that these be based on sound pseudo-random generators. This International Standard has been developed to help ensure that randomization, where needed, is carried out correctly and efficiently.

Six uses of randomization can be identified in statistical standardization:

- selection of a random sample;
- analysis of sample data;
- development of standards;
- checking theoretical results;
- demonstrating that a proposed procedure has the properties claimed of it;
- resolving uncertainty in the statistical literature.

# Random variate generation methods

## 1 Scope

This International Standard specifies methods for generating uniform and non-uniform random variates for Monte Carlo simulation purposes. Cryptographic random number generation methods are not included. This International Standard is applicable, *inter alia*, by

— researchers, industrial engineers or experts in operations management, who use statistical simulation,

— statisticians who need randomization related to SQC methods, statistical design of experiments or sample surveys,

— applied mathematicians who plan complex optimization procedures that require the use of Monte Carlo methods, and

— software engineers who implement algorithms for random variate generation.



## 2 Normative references



The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 2382-1, *Information technology — Vocabulary — Part 1: Fundamental terms*

ISO 3534-1, *Statistics — Vocabulary and symbols — Part 1: General statistical terms and terms used in probability*

ISO 3534-2, *Statistics — Vocabulary and symbols — Part 2: Applied statistics*

## 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 2382-1, ISO 3534-1 and ISO 3534-2 apply, except where redefined below.

**3.1**
**random variate**
**random number**
number as the realization of a specific random variable

NOTE 1 The term “random number” is often used for uniformly distributed random variate.

NOTE 2 Random numbers provided as a sequence are called a “random number sequence”.

**3.2**
**pseudo-random number**
**random number** (3.1) generated by an algorithm, that appears to be random

NOTE If there is no fear of misunderstanding, a pseudo-random number may simply be called a “random number”.

**3.3**
**physical random number**
**random number** (3.1) generated by a physical mechanism

**3.4**
**binary random number sequence**
**random number** (3.1) sequence consisting of zeros and ones

**3.5**
**seed**
initialization value required for pseudo-random number generation

## 4 Symbols and mathematical binary operations

### 4.1 Symbols

For the purposes of this document, the symbols given in the normative references as the latest versions of ISO/IEC 2382-1, ISO 3534-1 and ISO 3534-2 apply, except where redefined below.



The symbols and abbreviations specifically used in this International Standard are as follows:



$X$ integer type uniform random number

$U$ standard uniform random number

$Z$ normal random variate

$n$ suffix of random number sequence

### 4.2 Mathematical binary operations

The mathematical binary operations specifically used in this International Standard are as follows:

mod($m$; $k$) residue from dividing integer $m$ by $k$

$m \oplus k$ bitwise exclusive logical disjunction of binary integers $m$ and $k$

EXAMPLE 1 $1 \oplus 1 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$0 \oplus 0 = 0$

$1010 \oplus 1100 = 0110$

 

$m \wedge k$ bitwise logical conjunction of binary integers $m$ and $k$

EXAMPLE 2 $1 \wedge 1 = 1$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$0 \wedge 0 = 0$

$1010 \wedge 1100 = 1000$

$m := k$ replaces value $m$ by $k$

$m >> k$ $k$-bit right shift of binary integer $m$

$m << k$ $k$-bit left shift of binary integer $m$

# 5 Uniformly distributed pseudo-random numbers

## 5.1 General

This clause provides algorithms for generating uniformly distributed pseudo-random numbers based on M-sequence methods (see 5.2).



Annex A introduces the concept of physically generated random numbers for information.

Annex B includes C and full Basic codes for all the recommended algorithms for information. Although the linear congruential method is not recommended for complex Monte Carlo simulations, it is also included in Annex B for information.



## 5.2 M-sequence method definition

a) Let $p$ be a natural number, and $c_1, c_2, \ldots, c_{p-1}$ be specified to be 0 or 1, and define the recurrence formula

$$x_{n+p} = c_{p-1} x_{n+p-1} + c_{p-2} x_{n+p-2} + \ldots + c_1 x_{n+1} + x_n \pmod 2 \quad (n = 1, 2, 3, \ldots)$$

b) The least positive integer $N$ such that $x_{n+N} = x_n$ for all $n$ is called the period of the sequence. This sequence is called an M-sequence in cases where its period is $2^p - 1$.

c) The polynomial

$$t^p + c_{p-1} t^{p-1} + \ldots + c_1 t + 1$$

is called the characteristic polynomial of the above-mentioned recurrence formula.

NOTE 1 A necessary and sufficient condition for the above-mentioned recurrence formula to generate an M-sequence is that at least one of the seeds $x_1, x_2, \ldots, x_p$ is not zero.

NOTE 2 The letter M of the M-sequence originates from the English word “maximum”, which means the largest. The period of any sequence generated by the above recurrence formula cannot exceed $2^p - 1$. Therefore, if there is a series that has a period of $2^p - 1$, it is the series that has the largest period.

NOTE 3 When this method is used, either one of the polynomials listed in Table 1 or another primitive polynomial listed in the literature is chosen as the characteristic polynomial and its coefficients are used to define the recurrence formula in a).

## 5.3 Pentanomial GFSR method

This method uses a characteristic polynomial of 5 terms, and it generates binary integer sequences of $w$ bits by the following recurrence formula. This algorithm is called the GFSR or “generalized feedback shift register” random number generator.

$$X_{n+p} = X_{n+q1} \oplus X_{n+q2} \oplus X_{n+q3} \oplus X_n \quad (n = 1, 2, 3, \ldots)$$

The parameters are $(p, q1, q2, q3, w)$ and $X_1, \ldots, X_p$ are initially given as seeds. Examples of parameters $p$, $q_1$, $q_2$, $q_3$ giving the largest period $2^p - 1$ are indicated in Table 1.

**Table 1 — Pentanomial characteristic polynomials**

| $p$ | $q_1$ | $q_2$ | $q_3$ |
|---|---|---|---|
| 89 | 20 | 40 | 69 |
| 107 | 31 | 57 | 82 |
| 127 | 22 | 63 | 83 |
| 521 | 86 | 197 | 447 |
| 607 | 167 | 307 | 461 |
| 1 279 | 339 | 630 | 988 |
| 2 203 | 585 | 1 197 | 1 656 |
| 2 281 | 577 | 1 109 | 1 709 |
| 3 217 | 809 | 1 621 | 2 381 |
| 4 253 | 1 093 | 2 254 | 3 297 |
| 4 423 | 1 171 | 2 273 | 3 299 |
| 9 689 | 2 799 | 5 463 | 7 712 |
| NOTE $q_1$, $q_2$, $q_3$ represent exponents of the non-zero terms of the characteristic polynomial. | | | |



## 5.4 Combined Tausworthe method

Let $x_0, x_1, x_2, \ldots$ be an M-sequence generated by the recurrence relationship:

$$x_{n+p} = x_{n+q} + x_n \pmod 2 \quad (n = 0, 1, 2, \ldots)$$

Using this M-sequence, a $w$-bit integer sequence called a simple Tausworthe sequence with parameters $(p, q, t)$ is obtained as follows:

$$X_n = x_{nt} x_{nt+1} \ldots x_{nt+w-1} \quad (n = 0, 1, 2, \ldots)$$

where

$t$ is a natural number which is coprime to the period $2^p - 1$ of the M-sequence;

$w$ is the word length which does not exceed $p$.

The period of this sequence is also $2^p - 1$.

NOTE 1 Two integers are said to be coprime, or relatively prime, when they have no common divisors other than unity.

EXAMPLE If a primitive polynomial $t^4 + t + 1$ is chosen, set $p = 4$, and $q = 1$ in the above recurrence relationship. If the seeds $(x_0, x_1, x_2, x_3) = (1,1,1,1)$ are given to the recurrence, then the M-sequence obtained by the recurrence will be 1,1,1,1, 0,0,0,1, 0,0,1,1, 0,1,0,1, 1,1,1,0, … , and the period of the sequence is $2^4 - 1 = 15$. Taking, for example, $t = 4$ which is coprime to 15, and $w = 4$, the simple Tausworthe sequence $\{X_n\}$ with parameters (4, 1, 4) is obtained as follows:

$X_0 = x_0 x_1 x_2 x_3 = 1111$ (= 15)

$X_1 = x_4 x_5 x_6 x_7 = 0001$ (= 1)

$X_2 = x_8 x_9 x_{10} x_{11} = 0011$ (= 3)

$X_3 = x_{12} x_{13} x_{14} x_0 = 0101$ (= 5)

$X_4 = x_1 x_2 x_3 x_4 = 1110$ (= 14)

$X_5 = x_5 x_6 x_7 x_8 = 0010$ (= 2)

.....

The simple Tausworthe sequence obtained in this way will be, in decimal notation, 15, 1, 3, 5, 14, 2, 6, 11, 12, 4, 13, 7, 8, 9, 10, 15, 1, 3, … , and its period is $2^4 - 1 = 15$.

Suppose now that there is a multiple, say $J$, of simple Tausworthe sequences $\{X_n^{(j)}\}$, $j = 1, 2, ..., J$ with the same word length $w$. The combined Tausworthe method is a technique that generates a sequence of pseudo-random numbers $\{X_n\}$ as the bitwise exclusive logical disjunction in the binary representation of these $J$ sequences.



$$X_n = X^{(1)}{}_n \oplus X^{(2)}{}_n \oplus \dots \oplus X^{(J)}{}_n \quad (n = 0, 1, 2, \dots)$$



The parameters and the seeds of the combined Tausworthe sequence are combinations of the parameters and the seeds of each simple Tausworthe sequence. If the periods of the $J$ simple Tausworthe sequences are coprime, then the period of the combined Tausworthe sequence is the product of the periods of the $J$ sequences.

NOTE 2 This method can generate sequences with good multidimensional equidistribution characteristics. The algorithm taus88_31( ) given in Annex A generates a sequence of 31-bit integers by combining three simple Tausworthe generators with parameters $(p, q, t) = (31, 13, 12)$, (29, 2, 4), and (28, 3, 17), respectively. The period length of the combined sequence is $(2^{31} - 1)(2^{29} - 1)(2^{28} - 1)$, i.e. about $2^{88}$. Many other combinations are suggested in References [7] and [8] in the Bibliography.

## 5.5 Mersenne Twister method

Let $X_n$ be a binary integer of $w$ bits. Then, the Mersenne Twister method generates a sequence of binary integer pseudo-random numbers of $w$ bits according to the following recurrence formula with integer constants $p$, $q$, $r$ and a binary integer $a$ of $w$ bits.

$$X_{n+p} = X_{n+q} \oplus (X^{\mathrm{f}}{}_n | X^{\mathrm{l}}{}_{n+1})^{(r)}\,\mathbf{A}\,, \quad (n = 1, 2, 3, \dots)$$

where $(X^{\mathrm{f}}{}_n | X^{\mathrm{l}}{}_{n+1})^{(r)}$ represents a binary integer that is obtained by a concatenation of $X^{\mathrm{f}}{}_n$ and $X^{\mathrm{l}}{}_{n+1}$, the first $w - r$ bits of $X_n$ and the last $r$ bits of $X_{n+1}$ in this order. $\mathbf{A}$ is a $w \times w$ 0-1 matrix, which is determined by $a$, and the product $X\mathbf{A}$ is given by the following formula.

$X >> 1$ (when the last bit of $X = 0$)

$X\mathbf{A} = (X >> 1) \oplus a$ (when the last bit of $X = 1$)

Here, $X$ is regarded as a $w$ dimensional 0-1 vector.

 

NOTE The necessary amount of memory for this computation is $p$ words, the period becomes $2^{pw-r} - 1$, and the efficiency is better than that of the GFSR methods described previously. To improve the randomness of the first $w - r$ bits, the following series of conversions can be applied to $X_n$.

$$y := X_n$$

$$y := y \oplus (y >> u)$$

$$y := y \oplus [(y << s) \wedge b]$$

$$y := y \oplus [(y << t) \wedge c]$$

$$y := y \oplus (y >> l)$$

where $b$, $c$ are constant bits masks to improve the randomness of the first $w - r$ bits. The parameters of this algorithm are ($p$, $q$, $r$, $w$, $a$, $u$, $s$, $t$, $l$, $b$, $c$). The seeds are $X_2$, ..., $X_{q+1}$ and the first $w - r$ bits of $X_1$.

The final value of $y$ is the pseudo-random number.

# 6 Generation of random numbers from various distributions

## 6.1 Introduction



Methods of generating random numbers $Y$ from various distributions by using uniform random numbers $X$ of integer type, are described below.

The distribution function is denoted by $F(y)$. If it is a continuous distribution, its probability density function is denoted by $f(y)$, and if it is a discrete distribution, its probability mass function is denoted by $p(y)$.



## 6.2 Uniform distribution

### 6.2.1 Standard uniform distribution

#### 6.2.1.1 Probability density function

$$f(y) = \begin{cases} 1, & 0 \leqslant y \leqslant 1 \\ 0, & \text{otherwise} \end{cases}$$

#### 6.2.1.2 Random variate generation method

If the maximum value of uniform random number $X$ of integer type is $m - 1$, the following formula should be used to generate standard uniform random numbers.

$$U = \frac{X}{m}$$

EXAMPLE For any $w$-bit integer sequences generated by the method described in 5.2 through 5.5, $m = 2$.

NOTE 1 Because $X$ takes on discrete values, the values of $U$ are also discrete.

NOTE 2 The value of $U$ never becomes 1,0. The value of $U$ becomes 0,0 only when $X = 0$. In the case of M-sequence random numbers, any generation method may cause this phenomenon.

NOTE 3 Random numbers from the standard uniform distribution are called standard uniform random numbers, and are represented by $U_1$, $U_2$, … They are assumed to be independent of each other.

 

### 6.2.2 General uniform distribution

#### 6.2.2.1 Probability density function

$$f(y) = \begin{cases} 1/b, & a \leqslant y \leqslant a+b \\ 0, & \text{otherwise} \end{cases}$$

where $b > 0$.

#### 6.2.2.2 Random variate generation method

If the standard uniform random number $U$ is generated by the method specified in 6.2.1.2, then the general uniform random number should be generated by the following formula:

$$Y = bU + a$$

## 6.3 Standard beta distribution

### 6.3.1 Probability density function

$$f(y) = \begin{cases} \dfrac{y^{c-1}(1-y)^{d-1}}{\mathrm{B}(c,d)}, & 0 \leqslant y \leqslant 1 \\ 0, & \text{otherwise} \end{cases}$$



where $\mathrm{B}(c, d) = \int_0^1 x^{c-1}(1-x)^{d-1}\,\mathrm{d}x$ is the beta function and the parameters $c$ and $d$ are greater than 0.



### 6.3.2 Random variate generation method by Jöhnk

If the standard uniform random numbers $U_1$ and $U_2$ are independently generated by the method specified in 6.2.1, then the standard beta random number $Y$ should be generated by the following procedures.

If $\tilde{Y} = U_1^{1/c} + U_2^{1/d}$ is less than or equal to 1, set $Y = U_1^{1/c} / \tilde{Y}$; otherwise, generate two standard uniform random numbers again until the inequality is satisfied.

### 6.3.3 Random variate generation method by Cheng

If the standard uniform random numbers $U_1$ and $U_2$ are independently generated by the method specified in 6.2.1, then the standard beta random number $Y$ should be generated by the following procedures.

[Set-up]

a) Let

$$q = \begin{cases} \min(c, d), & \text{if } \min(c,d) < 1 \\ \sqrt{\dfrac{2cd - (c+d)}{c+d-2}}, & \text{otherwise} \end{cases}$$

[Generation]

b) Let

$$V = \frac{1}{q} \frac{U_1}{1-U_1},\ W = c\exp(V)$$

c) If

$$(c+d)\ln\left(\frac{c+d}{d+W}\right)+(c+q)V-\ln 4 \geqslant \ln(U_1^2U_2)$$

then employ

$$Y=\frac{W}{d+W}\text{; and stop.}$$

d) Generate $U_1$, $U_2$, and go to b).

$$V=\frac{1}{q}\frac{U_1}{1-U_1}(c+d)\ln\left(\frac{c+d}{d+W}\right)+(c+q)V-\ln 4 \geqslant \ln(U_1^2U_2)\,\frac{W}{d+W}$$

Jöhnk's method is recommended when max($c$, $d$) $\leqslant$ 1; otherwise, Cheng's method is recommended.

NOTE General beta random variates with the support [$a$, $a+b$] will be obtained by a linear transformation similar to the one described in 6.2.2.2.

## 6.4 Triangular distribution

### 6.4.1 Probability density function



$$f(y)=\begin{cases}\dfrac{b-|a-y|}{b^2}, & a-b\leqslant y\leqslant a+b\\ 0, & \text{otherwise}\end{cases}$$



where $b>0$.

### 6.4.2 Random variate generation method

If the standard uniform random numbers $U_1$ and $U_2$ are independently generated by the method specified in 6.2.1, then the triangular random number $Y$ should be generated by $Y=a+b(U_1+U_2-1)$.

## 6.5 General exponential distribution with location and scale parameters

### 6.5.1 Probability density function

$$f(y)=\begin{cases}\dfrac{1}{b}\exp\{-(y-a)/b\}, & y\geqslant a\\ 0, & y<a\end{cases}$$

where $a$ and $b$ are the location and scale parameters of the exponential distribution, respectively.

### 6.5.2 Random variate generation method

If the standard uniform random number $U$ is generated by the method specified in 6.2.1, then the general exponential random number should be generated by

$$Y=-\,b\ln(U)+a$$

## 6.6 Normal distribution

### 6.6.1 Probability density function

$$f(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{1}{2\sigma^2}(z-\mu)^2 \right\}, \quad -\infty < z < \infty$$

where $\mu$ and $\sigma$ are the mean and standard deviation of the normal distribution, respectively.

NOTE The symbol $Z$ is used for a normal random variate.

### 6.6.2 The Box-Müller method

If the standard uniform random numbers $U_1$ and $U_2$ are independently generated by the method specified in 6.2.1, then two independent normal random numbers $Z_1$, $Z_2$ will be generated by the following procedures:

$$Z_1 = \mu + \sigma\sqrt{-2\ln(1-U_1)}\cos(2\pi U_2)$$

$$Z_2 = \mu + \sigma\sqrt{-2\ln(1-U_1)}\sin(2\pi U_2)$$

NOTE 1 Since $U_1$ is not continuous, $Z_1$, $Z_2$ cannot be normally distributed in a strict sense. For example, using this procedure, the upper bound of the absolute value of the pseudo-standardized normal variates is $M = \sqrt{-2\ln(m^{-1})} = \sqrt{2\ln m}$; thus, when $m = 2^{32}$, $M \approx 6{,}660\,4$, and when $m = 2^{31} - 1$, $M \approx 6{,}555\,5$. However, since the probability that absolute values of random variables from a true standard normal distribution exceed $M$ is only about $10^{-10}$, this will rarely be a problem in practice.



NOTE 2 When generating $U_1$, $U_2$, by a linear congruential method sequentially, $U_1$ and $U_2$ are not independent of each other, so the tail of the distribution of the generated $Z_1$ and $Z_2$ can depart considerably from the true normal distribution.

## 6.7 Gamma distribution

### 6.7.1 Probability density function

$$f(y) = \begin{cases} \dfrac{1}{b\Gamma(c)}\{(y-a)/b\}^{c-1}\exp\{-(y-a)/b\}, & y \geqslant a \\ 0, & \text{otherwise} \end{cases}$$

where $a$, $b$, $c$ are the location, scale and shape parameters of the distribution, respectively.

### 6.7.2 Random variate generation methods

#### 6.7.2.1 General

Algorithms are given for three special cases depending on the shape parameter value $c$ as follows.

#### 6.7.2.2 Case of $c = k$ ($k$: integer)

Using independent uniform random numbers $U_1$, $U_2$, ... , $U_k$, the transformation formula

$$Y = a - b\ln\{(1-U_1)(1-U_2)\ldots(1-U_k)\}$$

should be used.